

Transform-domain Analysis of Packet Delay in Network Nodes with QoS-aware Scheduling

S. De Vuyst, S. Wittevrongel, and H. Bruneel

SMACS* Research Group, Department of Telecommunications and Information Processing, Ghent University, Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium.
`{sdv,sw,hb}@telin.ugent.be`

Abstract. In order to differentiate the perceived QoS between traffic classes in heterogeneous packet networks, equipment discriminates incoming packets based on their class, particularly in the way queued packets are scheduled for further transmission. We review a common stochastic modelling framework in which scheduling mechanisms can be evaluated, especially with regard to the resulting per-class delay distribution. For this, a discrete-time single-server queue is considered with two classes of packet arrivals, either delay-sensitive (1) or delay-tolerant (2). The steady-state analysis relies on the use of well-chosen supplementary variables and is mainly done in the transform domain. Secondly, we propose and analyse a *new* type of scheduling mechanism that allows precise control over the amount of delay differentiation between the classes. The idea is to introduce N reserved places in the queue, intended for future arrivals of class 1.

1 Introduction

In heterogeneous packet-based networks, the Quality of Service (QoS) perceived by a particular application highly depends on the presence of and interaction with *other* flows, due to statistical multiplexing and queueing in the intermediate network nodes. This is particularly the case in situations of high network load or congestion in intermediate network nodes. In a sense, managing a large heterogeneous network successfully, is a many-facetted problem, not unlike governing a society. Different groups of people have different characteristics, desires and aspirations, often conflicting with each other, but all of them are part of the same, highly interactive system. The actions of one group will always affect the others to some degree and it is the responsibility of the authorities to issue clear rules and laws as to how human interactions should take place, as well as to distribute the limited (monetary) resources among the different interest groups. Ideally, the primary aim of government's legislation and budget is to maximise prosperity and individual well-being. In this sense, traffic flows in a network are similar to people in society in many respects. Just like people, they share the same infrastructure, are unpredictable and may exhibit erratic or even malevolent behaviour. Therefore, enforcing strict rules and intelligent resource distribution are

* SMACS: Stochastic Modeling and Analysis of Communication Systems

required to achieve maximal QoS satisfaction. Hence, there is a continuing trend of introducing QoS-awareness into the equipment of network nodes, with the aim of sharing available network resources (e.g. bandwidth) among the traffic flows in a more intelligent and deliberate way, tailored as closely as possible to the needs of each flow. For this, many different techniques are used, both at the application level (traffic shaping, congestion notification) and at the level of the network (congestion avoidance, active queue management, specific packet scheduling and packet discarding mechanisms), see e.g. [25] for an overview. In answer to the scalability problems that arise when trying to meet the QoS demands of each separate flow in the network, the Differentiated Services (DiffServ) architecture [4] has been proposed. DiffServ is packet-based instead of flow-based. This means the individual flows are aggregated into a limited number of service classes and nodes decide on the per-hop behaviour of each packet based on its class. A simulation study of DiffServ with both delay and loss QoS classes is found in [44]. A *controllable* DiffServ mechanism is Proportional Differentiated Services [18], which aims to quantitatively differentiate the QoS among the classes relative to some predefined parameters. Many other scheduling mechanisms that limit the delay of a selected flow keep track of a ‘deadline’ for each queued packet, such as earliest deadline first (EDF) [43, 38] and alternative best-effort [26] or use virtual clocks [52]. An overview can be found in [25, 40]. Another approach to accommodate delay-sensitive traffic, is to keep queues small by discarding packets before congestion can arise. Several packet-discarding strategies have been presented and analysed in literature, such as push-out buffer (POB) [11], partial buffer sharing (PBS) [10, 27], random early detection (RED) [39, 23] and their variants. Aside from policies deciding whether or not a packet is admitted to the buffer, there is also a host of *scheduling mechanisms* that are responsible for determining which packet from the buffer is the next to be transmitted once the output link becomes available. QoS-*aware* scheduling mechanisms aim at providing somehow a better service to selected flows with higher importance, at the expense of a worse service delivered to the other flows. A theoretically ideal and fair way to share the server capacity over different flows is Generalised Processor Sharing (GPS) [41, 51], but this mechanism is difficult to apply in packet-based networks, so adaptations for packet scheduling are needed. For instance in the framework of ATM [3], weighted-round-robin (WRR) and weighted-fair-queueing (WFQ) [13] were proposed to achieve GPS-like weighted throughput and fairness [33]. For these mechanisms, there are separate queues for each type of traffic and the server ‘visits’ each queue in a cyclic, weighted and/or timed manner [42]. In the DiffServ framework, WFQ and WRR were recycled to ‘class-based’ WFQ and WRR, meaning that the flows (sometimes several thousands) are aggregated into a limited number of service classes between which a service differentiation is desirable.

We first review a common stochastic modelling framework by which the performance of some basic scheduling mechanisms can be evaluated, especially with regard to the resulting per-class delay distribution. A discrete-time single-server queueing model is considered with *two* classes (or ‘types’) of packet arrivals,

either *delay-sensitive* (type 1) or *delay-tolerant* (type 2). The queue operates in slotted time and all packets are assumed to require a single slot of service. We make abstraction of any packet loss and assume the queue to have infinite capacity. The arrival process is time-independent, although the numbers of 1- and 2-packets that arrive within a slot may be correlated. The equilibrium delay distribution experienced by each class is obtained via some standard intermediary steps. The first ① is to identify a minimal but sufficiently large state space by which the system can be described as a Markov process. As we will see, the number of required supplementary variables [12, 28] or ‘dimensions’ in the state space specifically depends on the considered scheduling mechanism. Secondly ② the equilibrium distribution of the system’s state over this space is calculated as a joint probability generating function (pgf), which in turn is required for the final step, ③ obtaining the per-class delay distribution, also in the form of a generating function. This *generating-functions* approach, also outlined in e.g. [29, 49, 5], has proven to be successful in application to an impressively broad range of queueing models. We here specifically apply it to assess the performance of an alternative scheduling mechanism called the MR (*Multiple Reservation*) mechanism, which we introduce in the next section.

2 Class-based scheduling mechanisms

The quintessential QoS-flat scheduling mechanism (or ‘queueing discipline’) is FIFO (First-In First-Out) where packets are served in their arrival order, without regard of their class. Occasionally useful disciplines are also LIFO (Last-In First-Out) [2] and ROS (Random Order of Service) [30, 22]. The latter not only neglects the class but also the arrival order of the packets. Clearly, no significant differentiation in the per-class delay distribution is expected under any of these disciplines. On the other hand, the most *extreme* way of service differentiation is AP (Absolute Priority) or ‘HOL priority’ (Head of Line), either preemptive or non-preemptive, see [45, 47, 24, 9, 48] as well as the contribution of Walraevens et al. in this volume. Under AP, the next scheduled packet is the one that (1) belongs to the set of queued packets with highest priority and (2) has the longest waiting time of the packets in the set. If there are M traffic classes with lower classes having higher priority, then a class- i packet only has a transmission opportunity in the complete absence of packets from classes 1 to $i-1$. Hence, AP guarantees the *lowest* possible delay for class-1 traffic, at the cost of increasing the delay for the other classes. The drawback is that classes with lower priority may experience dramatically high delay, especially when the partial load of the higher-priority classes is high, an effect known as *packet starvation*. Therefore, AP may be a bridge too far in many situations. Several amendments to AP have been proposed that try to soften the severe strictness of the policy. For example, in [46] a Probabilistic Priority scheme is discussed that assigns a small probability p_i to each class by which the server may skip service to class $i+1$, even though class- i packets are available. In case of two priority classes, [35, 36] analyse different ways of implementing priority jumps (PJ), as proposed in [32].

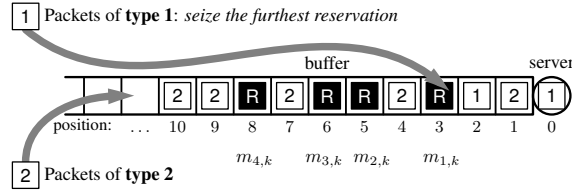


Fig. 1. A queue operating under MR with $N = 4$. The queue spaces are numbered as indicated, and the positions of the reservations at the start of slot k are given by the variables $m_{j,k}$ ($j = 1, \dots, N$).

The idea is to allow class-2 packets (with lower priority) to promote to class 1 on some occasions and ‘jump’ to the tail of the high-priority queue. Considered jumping schemes are: HOL-PJ where class-2 packets jump after some delay threshold, HOL-MBP (Merge-by-probability) where the *entire* class-2 queue jumps with some probability β , HOL-JOS (Jump-or-serve) where jumps occur at the same rate as the output link, HOL-JIA (Jump-if-Arrival) where jumps are synchronised with class-2 arrivals, and so on. We here specifically focus on the performance analysis of a new queue scheduling mechanism that can realise a statistic reduction of the delay for type 1 traffic at the cost of increasing the delay for type 2 traffic. This delay differentiation is achieved by introducing a total of N ($N \geq 1$) *reserved spaces* in the queue that will be occupied by future arriving packets of type 1. We refer to this scheduling mechanism as the *Multiple Reservation* (MR) discipline. The basic idea was first coined by Burakowski and Tarasiuk in [8], where a rudimentary estimation of the mean delay for both traffic types is given in case of Poisson arrival flows. In our previous work [14, 15], we have obtained exact expressions for the equilibrium distribution of the delay of both packet types, in case of a single reserved position (i.e. $N = 1$), as well as for geometric service times in [19]. This was subsequently extended in [16, 17] to the general case $N > 1$ where efficient numerical procedures were presented to calculate the pgf, mean and tail distribution of the delay experienced by both types of packets. As a convention, let us number the positions of the spaces in the queue as in Fig. 1, i.e. the server has position 0, while the queue spaces have numbers that increase with their distance to the server. The policy for storing and scheduling packets in the queue is then the following:

- Initially, there are N reserved positions (R’s) in the queue, on positions 1 to N . This is also the case every time the queue is empty or ‘idle’.
- Of all the packets arriving in a slot, the 1-packets are stored in the queue first and then the 2-packets.
- If a 1-packet is stored, it seizes the most advanced R in the queue (i.e. the one with the lowest position number) and makes a *new* R at the end of the queue. A 2-packet on the other hand, is stored at the end of the queue in the usual FIFO manner.

- As long as it is not seized by a 1-packet, an R behaves as a normal packet in the sense that it shifts one place to the right every time a packet leaves the server. However, an R cannot enter the server at position 0, nor can it leave the queue.

It is clear that the number of R's always remains equal to N , due to the fact that each 1-packet seizes an R but at the same time also creates an R at the end of the queue. Note that MR is work-conserving and maintains the per-type packet ordering, just like FIFO and AP. The differentiation in delay arises from the fact that 1-packets may jump over already stored 2-packets. For example, in the situation of Fig. 1, the next arriving 1-packet will take position 3, thereby jumping over 4 packets of type 2. Clearly, the higher we choose N the more the delays of type 1 and type 2 packets will differ. More so, if $N \rightarrow \infty$ the queue operates in exactly the same way as AP, providing the maximum possible delay differentiation. In this way, the number of reservations N in the queue can be seen as a parameter by which the *amount* of delay differentiation can be carefully controlled.

3 Discrete-time queueing model

In the following, we propose a discrete-time queueing model by which the per-class delay distribution under a specific scheduling mechanism can be assessed quantitatively. We assume all packets are stored in a single queue with infinite capacity and that each packet requires one slot service. In general, packets belong to one of M service classes with class i having more strict delay requirements (higher priority) than class $i+1$. Let $a_{i,k}$ denote the number of packets of type i ($1 \leq i \leq M$) that arrive in the queue during slot k . We assume that the numbers of arrivals in subsequent slots are independent and identically distributed (iid), with their joint distribution given by the pgf

$$\mathcal{A}(z_1, \dots, z_M) \triangleq \mathbb{E}[z_1^{a_{1,k}} \dots z_M^{a_{M,k}}]. \quad (1)$$

This allows for the numbers of arrivals of different classes during the same slot to be correlated. We assume *delayed access* for the packets, meaning that any arrivals during slot k are not stored in the queue until the end of that slot. Consequently, no packet can be served during the slot in which it arrived. Let the random variable $v_{i,k}$ denote the number of packets of type i present in the system at the *beginning* of slot k , summing up to $u_k = v_{1,k} + \dots + v_{M,k}$, denoting the total system content at that time. In case of QoS-flat disciplines, it is often logical to single out a traffic class i and regard the other as an aggregated class \bar{i} . Let $a_{\bar{i},k} = \sum_{j \neq i} a_{j,k}$, then $A(x, z) \triangleq \mathbb{E}[x^{a_{\bar{i},k}} z^{a_{i,k}}] = \mathcal{A}(x, \dots, x_{i-1}, z, x_{i+1}, \dots, x_M)$ is the joint distribution of the arrivals in slot k of classes \bar{i} and i . This is equivalent to considering $M = 2$, with type-1 packets having delay priority over type-2 packets, and joint pgf $A(z_1, z_2) = \mathbb{E}[z_1^{a_{1,k}} z_2^{a_{2,k}}]$. We will also use the marginal distributions $A_i(z)$ of the number of arrivals per slot of type $i = 1, 2$, which are given by

$$A_1(z) \triangleq \mathbb{E}[z^{a_{1,k}}] = A(z, 1), \quad A_2(z) \triangleq \mathbb{E}[z^{a_{2,k}}] = A(1, z). \quad (2)$$

The *total* number of arrivals (both of type 1 and 2) during slot k is denoted by $a_{T,k} = a_{1,k} + a_{2,k}$ with pgf $A_T(z) = E[z^{a_{T,k}}] = A(z, z)$. The mean number of arrivals per slot (arrival rate) of type i follows from the moment-generating property of pgfs as $\lambda_i = E[a_{i,k}] = A'_i(1)$, while the total arrival rate is $\lambda_T = \lambda_1 + \lambda_2 = A'_T(1)$. Some higher-order derivatives evaluated for $z = 1$ are $\lambda'_i = A''_i(1)$ and $\lambda'_T = A''_T(1)$, as well as the mixed moment $\lambda_{12} = \frac{\partial^2}{\partial z_1 \partial z_2} A(1, 1) = E[a_1 a_2]$. To keep the forthcoming expressions tractable, we use the following notation for the distribution of the type-1 arrivals in the probability domain. Let the mass function of the number of 1-arrivals per slot be

$$\beta_i \triangleq \text{Prob}[a_{1,k} = i], \quad i \geq 0, \quad \text{and} \quad \alpha \triangleq \beta_0 = A_1(0) = A(0, 1), \quad (3)$$

such that $A_1(z) = \sum_{i=0}^{+\infty} \beta_i z^i$. Additionally, define

$$A_{i*}(z) \triangleq E[z^{a_{2,k}} \{a_{1,k} = i\}], \quad i \geq 0, \quad (4)$$

which is the pgf of the number of 2-arrivals in case there are exactly i arrivals of type 1 in the same slot, and where $E[X\{Y\}] = E[X|Y] \text{Prob}[Y]$. Note also that $\beta_i = A_{i*}(1)$, $i \geq 0$.

4 Equilibrium state of multi-class queues

The ‘system’ we are considering is the queue as specified above, containing packets of both classes. Our final goal is to obtain the distribution of the per-class delay. Formally, let us consider an *arbitrary* class- i packet ($i = 1, 2$) and tag it as packet \mathcal{P} . Denote the arrival slot of \mathcal{P} as slot I . We define the delay d_i of \mathcal{P} as the number of slots between the end of slot I and the end of the slot in which \mathcal{P} departs from the queue. It is clear that for most scheduling disciplines, d_i heavily depends on the state of the system when \mathcal{P} arrives. For example, if the system content u_I is large at the moment \mathcal{P} arrives, its delay will typically be large as well, because all of the already present packets have to be served prior to \mathcal{P} ’s transmission. This is the reason why intermediate step ② of obtaining the exact equilibrium system state is required.

As mentioned in section 1, the first step ① however is to identify a suitable description of the system’s state during an arbitrary slot k which contains enough information so that the per-class delay distribution can be derived from it. The variables $\langle s_1, \dots, s_n \rangle$ that are chosen to be part of the system state are called the *system state variables* and we characterise their distribution in slot k by their joint pgf P_k . The primary concern here, is to choose the collection of system state variables in such a way that it is Markovian, in the sense that P_{k+1} can be determined from P_k without relying on any information pertaining to slots other than k . We assume that the system reaches equilibrium for $k \rightarrow \infty$, i.e. that appropriate stability conditions for the system’s parameters are met. Whenever we deal with random variables assumed in this equilibrium regime, we may drop the time index k where appropriate. For the equilibrium distribution of the system state we thus write $P = \lim_{k \rightarrow \infty} P_k$. For a general arrival process,

note however that the distribution P_I of the system state in the arrival slot of \mathcal{P} may be *different* from the overall equilibrium distribution P . Nevertheless, it is known [5] that due to the temporally uncorrelated (iid) nature of the packet arrival process we assumed in section 3, the system state as ‘seen’ by an *arbitrary arriving packet* of either type has the same distribution as the system state in an *arbitrary slot*, i.e. $P_I = P$. The results we obtain in this section can therefore immediately be used for the delay analysis in sections 5 and 6.

So let us apply steps ① and ② to the QoS-flat disciplines first. For 2-class FIFO, it suffices to maintain the total system content $\langle u_k \rangle$ as per-slot information, i.e. the only required system state variable is u_k . The evolution of the system content is governed by the well-known Lindley equation:

$$u_{k+1} = (u_k - 1)^+ + a_{T,k}, \quad (5)$$

where $(\cdot)^+$ is the operator $\max(\cdot, 0)$. The equilibrium pgf of the system state then follows as

$$U(z) \triangleq \lim_{k \rightarrow \infty} E[z^{u_k}] = A_T(z) E[z^{(u-1)^+}] = (1 - \lambda_T) \frac{A_T(z)(z - 1)}{z - A_T(z)}, \quad (6)$$

see [5, 6, 29]. Applying the moment-generating property of pgfs on (6), as well as applying de l’Hôpital’s rule, we obtain the mean equilibrium system content as

$$E[u] = U'(1) = \lambda_T + \frac{\lambda'_T}{2(1 - \lambda_T)}. \quad (7)$$

The importance of (6) and (7) is the fact that they hold for most work-conserving disciplines. The *total* system content u , i.e. *all* packets irrespective of their class, is distributed as in (6) for all the scheduling mechanisms we consider in this paper. Whether the total system content alone suffices as system state description is another matter of course. For LIFO, no system state needs to be maintained at all (i.e. $\langle \rangle$) since the delay of \mathcal{P} is independent of the number of packets u_I already stored in the system. For ROS on the other hand, $\langle u_k \rangle$ is again sufficient.

For QoS-aware scheduling mechanisms, the delay of \mathcal{P} usually depends on how many of the present packets in the system are of type 1 and how many of type 2. In case of AP [47] however, it turns out that keeping track of the total system content u_k alone is sufficient for the delay analysis, i.e. we require $\langle u_k \rangle$, leading again to (5)–(7). Nevertheless, the joint pgf of v_1 and v_2 can be obtained using the system equations

$$v_{1,k+1} = (v_{1,k} - 1)^+ + a_{1,k}, \quad v_{2,k+1} = \begin{cases} (v_{2,k} - 1)^+ + a_{2,k} & \text{if } v_{1,k} = 0, \\ v_{2,k} + a_{2,k} & \text{if } v_{1,k} > 0. \end{cases} \quad (8)$$

The joint pgf of v_1 and v_2 then found as [47]

$$U(z_1, z_2) = E[z_1^{v_1} z_2^{v_2}] = (1 - \lambda_T) \frac{A(z_1, z_2)(z_2 - 1)}{z_1 - A(z_1, z_2)} \frac{(z_1 - X(z_2))}{z_2 - X(z_2)}, \quad (9)$$

where the function $X(z)$ is implicitly defined by $X(z) = A(X(z), z)$ and $X(1) = 1$. For the variants of mechanisms with priority jumps (PJ) [35, 36] mentioned in the introduction however, the required state space is usually $\langle v_{1,k}, v_{2,k} \rangle$, leading to different variants of (9).

For the Multiple Reservation mechanism (MR), recall that the j th packet in an arriving batch of 1-packets seizes the j th reserved space if $j \leq N$. Hence, besides the queue content u_k we also need to keep track of the precise positions of all N reservations in the queue. We number the R's from 1 to N , with the first reservation being the one *closest* to the server, i.e. the one with the smallest position number. Conversely, we may also say that the j th reservation ($j = 1, \dots, N$) has *order* j . At the start of slot k , we denote the position of the order j reservation by $m_{j,k}$, as is illustrated in Fig. 1. As a 1-packet always seizes the R with the lowest position number, it is seen that there can be no 1-packets in the queue on positions larger than $m_{1,k}$. In other words, all packets behind the first reservation must be of type 2. So what variables need to be added to the system state in order for the per-class delay distribution to be derived? If \mathcal{P} is of type 1 and the j th 1-packet of an arriving batch, it will seize the j th order reservation ($j \leq N$). Each time, a new reservation is made at the end of the queue, at position $u_k + N$. Therefore, it is clear that the positions of all R's together with the queue content u_k need to be part of the system's state variables, i.e. $\langle m_{j,k}, j=1, \dots, N; u_k \rangle$.

The following system equations establish the value of the system state variables in slot $k+1$, for all possible values of those variables in slot k . The working method is to start from a certain state at the start of slot k . Then consider every possible event *during* this slot in terms of arrivals, storage, scheduling and departures and finally, write down the new system state this results in at the start of slot $k+1$. In principle, this yields a function from one $(N+1)$ -dimensional space to another, although this space can be somewhat reduced by ruling out states than can never be reached. For instance, because of their physical meaning, we know that the system state variables must satisfy the constraint

$$1 \leq m_{1,k} < m_{2,k} < \dots < m_{N,k} \leq (u_k - 1)^+ + N, \quad (10)$$

which for the position of the j th reservation individually results in

$$j \leq m_{j,k} \leq (u_k - 1)^+ + j, \quad j=1, \dots, N. \quad (11)$$

As a convention, let j indicate any value from 1 to N , unless stated otherwise. In our analysis it turns out that, instead of the variables $m_{j,k}$, it is often more convenient to work with the variables

$$\hat{m}_{j,k} \triangleq m_{j,k} - j, \quad (12)$$

which all have 0 as their minimal value instead of j . Therefore, the constraints (10) and (11) now respectively become

$$0 \leq \hat{m}_{1,k} \leq \hat{m}_{2,k} \leq \dots \leq \hat{m}_{N,k} \leq (u_k - 1)^+, \quad (13)$$

$$0 \leq \hat{m}_{j,k} \leq (u_k - 1)^+ . \quad (14)$$

Obviously, knowledge of the value or distribution of $m_{j,k}$ implies that of $\hat{m}_{j,k}$ and vice versa, so we may interchangeably use both as system state variables.

For the system equations we can distinguish between four cases, in all of which the new system content u_{k+1} is determined by (5). Observe also that $a_{2,k}$ appears in (5) but not in any of the following equations for $\hat{m}_{j,k+1}$ where only the number of arrivals of type 1 is of importance. Assuming that the system is not empty to begin with, the events during slot k can generally be summarised as follows. First, there are $a_{1,k}$ 1-arrivals to be stored. One by one they seize the first R they see and make a new one at the end. As such, seen as a group, the first N of these 1-arrivals take R's that existed before slot k , while any remaining 1-arrivals seize an R that was created by a previous 1-arrival in slot k . At the end of the slot, after the 2-packets have been stored as well, the packet in the server terminates its service and leaves the queue. Then, at the start of slot $k+1$ a new packet will enter service, at least if there are any left in the system. It is the first packet (the one with lowest position number p) that will jump over any R's at positions 1 to $p-1$ into the server at position 0. Then, since position p is free now, all packets and reservations on positions larger than p shift one position towards the server. As we have said, these considerations lead us to distinguish four groups of system equations as follows.

► $u_k = 0$ (empty system)

First of all, in case of an empty system we know that the N reservations are grouped together on positions 1 to N , so

$$u_k = 0 \quad \Rightarrow \quad \hat{m}_{1,k} = \hat{m}_{2,k} = \dots = \hat{m}_{N,k} = 0 .$$

Therefore, we have in slot $k+1$:

$$\hat{m}_{j,k+1} = (a_{1,k} - 1)^+ . \quad \textbf{(Empty)}$$

► $u_k > 0, a_{1,k} = 0$ (no 1-arrivals)

In this and the remaining cases, we know that the system is not empty in slot k . Since $a_{1,k} = 0$ there are no arrivals of type 1 here. None of the reservations will be seized so they all survive to the next slot. However, after the packet in service during slot k has left, they will be shifted by one position as far as the lower constraint in (14) is not violated. We have

$$\hat{m}_{j,k+1} = (\hat{m}_{j,k} - 1)^+ . \quad \textbf{(Keep)}$$

► $u_k > 0, a_{1,k} = i$ with $1 \leq i < N$

In this case, the number of 1-arrivals i is smaller than the number of reservations N . These i arrivals seize the first i reservations, i.e. those at positions $m_{1,k}$ up to $m_{i,k}$ and make i new reservations at the end of the queue. So the *last* i reservations in the next slot will be newly created and positioned together at the end of the queue. Then, accounting for the fact that one packet will leave, it turns out that we have

$$\hat{m}_{j,k+1} = u_k + i - 2, \quad \text{if } j = N - i + 1, \dots, N . \quad \textbf{(AtEnd)}$$

On the other hand, the *first* $N-i$ R's in the new slot are reservations that were not seized and have survived. Their ordering number has simply decreased by i or in other words, they have ' i -shifted':

$$\hat{m}_{j,k+1} = \hat{m}_{j+i,k} + i - 1, \quad \text{if } j = 1, \dots, N-i. \quad (\textbf{i-shift})$$

► $u_k > 0$, $a_{1,k} = i$ with $i \geq N$

Now, there are at least as many 1-arrivals as there are reservations. In this case all new reservations will be grouped at the end of the queue, since none of the old R's survive. Equation (**AtEnd**) now applies for *all* new reservation positions, i.e.

$$\hat{m}_{j,k+1} = u_k + i - 2, \quad j = 1, \dots, N. \quad (\textbf{AtEnd})$$

Now we use the above equations to obtain the equilibrium distribution of the system state $\langle m_{j,k}, j = 1, \dots, N; u_k \rangle$, of which we define the joint pgf as

$$P_k(y_1, y_2, \dots, y_N; z) \triangleq E[y_1^{\hat{m}_{1,k}} y_2^{\hat{m}_{2,k}} \dots y_N^{\hat{m}_{N,k}} z^{u_k}]. \quad (15)$$

The equations (**Empty**), (**Keep**), (**i-shift**) and (**AtEnd**) allow to relate the joint pgf P_{k+1} of the system state in slot $k+1$ to the distribution P_k in slot k . We perform separate calculations for the four cases above, splitting up the joint pgf into four terms as

$$\begin{aligned} P_{k+1}(y_1, y_2, \dots, y_N; z) &= E[y_1^{\hat{m}_{1,k+1}} y_2^{\hat{m}_{2,k+1}} \dots y_N^{\hat{m}_{N,k+1}} z^{u_{k+1}}] \\ &= E[\dots \{u_k = 0\}] + E[\dots \{u_k > 0, a_{1,k} = 0\}] \\ &\quad + \sum_{i=1}^{N-1} E[\dots \{u_k > 0, a_{1,k} = i\}] + \sum_{i=N}^{+\infty} E[\dots \{u_k > 0, a_{1,k} = i\}]. \end{aligned} \quad (16)$$

For the first term, (**Empty**) applies, as well as (5) such that

$$\begin{aligned} &E[y_1^{\hat{m}_{1,k+1}} y_2^{\hat{m}_{2,k+1}} \dots y_N^{\hat{m}_{N,k+1}} z^{u_{k+1}} \{u_k = 0\}] \\ &= E[(y_1 y_2 \dots y_N)^{(a_{1,k}-1)^+} z^{a_{1,k}+a_{2,k}} \{u_k = 0\}] \\ &= \dots \\ &= \frac{p_{0,k}}{y_1 y_2 \dots y_N} [(y_1 y_2 \dots y_N - 1)A(0, z) + A(y_1 y_2 \dots y_N z, z)], \end{aligned} \quad (17)$$

where, $p_{0,k} = U_k(0) = P_k(0, 0, \dots, 0; 0)$ is the probability that the system is empty at the beginning of slot k . The second term of (16) can be further developed with (**Keep**), which yields

$$\begin{aligned} &E[y_1^{\hat{m}_{1,k+1}} y_2^{\hat{m}_{2,k+1}} \dots y_N^{\hat{m}_{N,k+1}} z^{u_{k+1}} \{u_k > 0, a_{1,k} = 0\}] \\ &= E[y_1^{(\hat{m}_{1,k}-1)^+} y_2^{(\hat{m}_{2,k}-1)^+} \dots y_N^{(\hat{m}_{N,k}-1)^+} z^{u_k-1+a_{2,k}} \{u_k > 0, a_{1,k} = 0\}] \\ &= A(0, z) E[y_1^{(\hat{m}_{1,k}-1)^+} y_2^{(\hat{m}_{2,k}-1)^+} \dots y_N^{(\hat{m}_{N,k}-1)^+} z^{u_k-1} \{u_k > 0\}] \end{aligned}$$

$$\begin{aligned}
&= A(0, z) \sum_{n=1}^{+\infty} \sum_{j_1=0}^{n-1} \sum_{j_2=j_1}^{n-1} \sum_{j_3=j_2}^{n-1} \cdots \sum_{j_N=j_{N-1}}^{n-1} y_1^{(j_1-1)^+} \cdots y_N^{(j_N-1)^+} z^{n-1} p_k(j_1, \dots, j_N; n) \\
&= A(0, z) \sum_{n=1}^{+\infty} z^{n-1} \left[\begin{aligned}
&\sum_{j_1=1}^{n-1} \sum_{j_2=j_1}^{n-1} \sum_{j_3=j_2}^{n-1} \sum_{j_4=j_3}^{n-1} \cdots \sum_{j_N=j_{N-1}}^{n-1} y_1^{j_1-1} y_2^{j_2-1} y_3^{j_3-1} \cdots y_N^{j_N-1} p_k(j_1, \dots, j_N; n) \\
&+ \sum_{j_2=1}^{n-1} \sum_{j_3=j_2}^{n-1} \sum_{j_4=j_3}^{n-1} \cdots \sum_{j_N=j_{N-1}}^{n-1} y_2^{j_2-1} y_3^{j_3-1} \cdots y_N^{j_N-1} p_k(0, j_2, \dots, j_N; n) \\
&+ \sum_{j_3=1}^{n-1} \sum_{j_4=j_3}^{n-1} \cdots \sum_{j_N=j_{N-1}}^{n-1} y_3^{j_3-1} \cdots y_N^{j_N-1} p_k(0, 0, j_3, \dots, j_N; n) \\
&+ \dots \\
&+ \sum_{j_N=1}^{n-1} y_N^{j_N-1} p_k(0, 0, \dots, 0, j_N; n) \\
&+ p_k(0, 0, \dots, 0; n) \end{aligned} \right] \\
&= \frac{A(0, z)}{z} \left[\begin{aligned}
&\frac{1}{y_1 y_2 y_3 \cdots y_N} \left(P_k(y_1, y_2, y_3, \dots, y_N; z) - P_k(0, y_2, y_3, \dots, y_N; z) \right) \\
&+ \frac{1}{y_2 y_3 \cdots y_N} \left(P_k(0, y_2, y_3, \dots, y_N; z) - P_k(0, 0, y_3, \dots, y_N; z) \right) \\
&+ \dots \\
&+ \frac{1}{y_N} \left(P_k(0, 0, \dots, 0, y_N; z) - P_k(0, 0, \dots, 0, 0; z) \right) \\
&+ P_k(0, 0, \dots, 0, 0; z) - p_{0,k} \end{aligned} \right], \tag{18}
\end{aligned}$$

where we have used the following notation for the mass function of the system state distribution in slot k :

$$p_k(j_1, j_2, \dots, j_N; n) \triangleq \text{Prob}[\hat{m}_{1,k}=j_1, \hat{m}_{2,k}=j_2, \dots, \hat{m}_{N,k}=j_N, u_k=n]. \tag{19}$$

In the third term of (16), we must apply **(i-shift)** for the new reservations of order 1 to $N-i$ and **(AtEnd)** for the remaining reservation positions. We have for $1 \leq i < N$, using (4):

$$\begin{aligned}
&\mathbb{E} \left[\underbrace{y_1^{\hat{m}_{1,k+1}} \cdots y_{N-i}^{\hat{m}_{N-i,k+1}}}_{(i\text{-shift})} \cdot \underbrace{y_{N-i+1}^{\hat{m}_{N-i+1,k+1}} \cdots y_N^{\hat{m}_{N,k+1}}}_{(\text{AtEnd})} \cdot z^{u_{k+1}} \{u_k > 0, a_{1,k} = i\} \right] \\
&= \mathbb{E} [y_1^{\hat{m}_{i+1,k}+i-1} \cdots y_{N-i}^{\hat{m}_{N,k}+i-1} y_{N-i+1}^{u_k+i-2} \cdots y_N^{u_k+i-2} z^{u_k-1+i+a_{2,k}} \{u_k > 0, a_{1,k} = i\}] \\
&= (zy_1 y_2 \cdots y_{N-i})^{i-1} (y_{N-i+1} \cdots y_N)^{i-2} A_{i*}(z)
\end{aligned}$$

$$\begin{aligned}
& \mathbb{E}[y_1^{\hat{m}_{i+1,k}} \dots y_{N-i}^{\hat{m}_{N,k}} (zy_{N-i+1} \dots y_N)^{u_k} \{u_k > 0\}] \\
&= (zy_1 y_2 \dots y_{N-i})^{i-1} (y_{N-i+1} \dots y_N)^{i-2} A_{i*}(z) \\
& \quad [P_k(1, 1, \dots, \underset{i}{1}, y_1, y_2, \dots, y_{N-i}; zy_{N-i+1} \dots y_N) - p_{0,k}]. \tag{20}
\end{aligned}$$

Finally, in the last term of (16), we find for $i \geq N$ using **(AtEnd)**,

$$\begin{aligned}
& \mathbb{E}[y_1^{\hat{m}_{1,k+1}} y_2^{\hat{m}_{2,k+1}} \dots y_N^{\hat{m}_{N,k+1}} z^{u_{k+1}} \{u_k > 0, a_{1,k} = i\}] \\
&= z^{i-1} (y_1 y_2 \dots y_N)^{i-2} A_{i*}(z) [U_k(zy_1 y_2 \dots y_N) - p_{0,k}]. \tag{21}
\end{aligned}$$

Adding up the terms (17), (18), (20) and (21), we get the right-hand side of (16). If equilibrium kicks in, we can single out the function $P(y_1, \dots, y_N; z)$ and find our basic functional equation for the equilibrium distribution of the MR system state with N reservations:

$$\begin{aligned}
& (z \tilde{y}_1 - A(0, z)) P(y_1, y_2, \dots, y_N; z) \\
&= z p_0 \left[(\tilde{y}_1 - 1) A(0, z) + A(z \tilde{y}_1, z) \right] \\
& \quad + A(0, z) \left[(y_1 - 1) P(0, y_2, \dots, y_N; z) \right. \\
& \quad \quad + y_1 (y_2 - 1) P(0, 0, y_3, \dots, y_N; z) \\
& \quad \quad + y_1 y_2 (y_3 - 1) P(0, 0, 0, y_4, \dots, y_N; z) \\
& \quad \quad + \dots \\
& \quad \quad \left. + y_1 y_2 \dots y_{N-1} (y_N - 1) P(0, 0, \dots, 0; z) - \tilde{y}_1 p_0 \right] \\
& \quad + \sum_{i=1}^{N-1} \frac{(z \tilde{y}_1)^i}{\tilde{y}_{N-i+1}} A_{i*}(z) \left[P(1, \dots, \underset{i}{1}, y_1, y_2, \dots, y_{N-i}; z \tilde{y}_{N-i+1}) - p_0 \right] \\
& \quad + \sum_{i=N}^{+\infty} z^i \tilde{y}_1^{i-1} A_{i*}(z) \left[U(z \tilde{y}_1) - p_0 \right], \tag{22}
\end{aligned}$$

where we define \tilde{y}_j as the product $y_j y_{j+1} \dots y_N$. This functional equation completely determines the equilibrium distribution P , although we see that a lot of unknown functions have yet to be determined. Nevertheless, all these unknowns can be resolved by using relation (22) only, as we will demonstrate. Observe that there are a total of $2N$ unknown functions on the right-hand side of (22). Let us designate a shorthand to each of these functions and order them in a list as follows

1. $\boxed{1} = P(0, y_2, y_3, y_4, \dots, y_N; z)$
2. $\boxed{1} = P(1, y_2, y_3, y_4, \dots, y_N; z)$
3. $\boxed{2} = P(0, 0, y_3, y_4, \dots, y_N; z)$
4. $\boxed{2} = P(1, 1, y_3, y_4, \dots, y_N; z)$
5. $\boxed{3} = P(0, 0, 0, y_4, \dots, y_N; z)$
6. $\boxed{3} = P(1, 1, 1, y_4, \dots, y_N; z)$

$$\begin{aligned}
& \vdots \\
2N-1. \quad \mathbb{N} &= P(0, 0, 0, 0, \dots, 0; z) \\
2N. \quad \mathbb{N} &= P(1, 1, 1, 1, \dots, 1; z) = U(z)
\end{aligned} \tag{23}$$

Note that we could have included the probability p_0 in this list as well, although it easily follows as $p_0 = 1 - \lambda_T$ by imposing the normalisation condition on $U(z)$. The unknown functions can be determined in this order by performing the appropriate substitutions in (22). In fact, the functional equation is able to provide each of the unknowns in the list as a function of unknowns *further* in the list. To make clear how this is done, let us denote by \blacksquare the function $P(y_1, y_2, \dots, y_N; z)$ in an *explicit* form, i.e. equal to (22) but with all unknowns (those in list (23)) resolved. On the other hand, we represent by $\blacksquare ? \mathbb{1}\mathbb{1}\mathbb{2}\mathbb{2} \dots \mathbb{N}\mathbb{N}$ a relation determining $P(y_1, y_2, \dots, y_N; z)$, but in which the functions after the question mark are still unresolved. Obviously, this is the functional equation in the form given by (22). Clearly, the final explicit expression for the equilibrium distribution of the system state we are looking for is \blacksquare . Even for small N though, obtaining \blacksquare is an enormous task to do by hand, so we only explain *how* to do this, rather than actually doing it.

As we have said, (22) holds the key to determining all the unknown functions explicitly by evaluating it for the right arguments. In what follows, we describe a ‘binary tree backtracking’ scheme that shows us the way. There are *two types* of substitutions that yield relevant information. The first one is to let

$$y_1 \rightarrow 1, y_2 \rightarrow 1, \dots, y_{n-1} \rightarrow 1, y_n \rightarrow 1, \tag{24}$$

for some $n = 1, \dots, N$. This directly gives the relation $\blacksquare ? \mathbb{n+1}\mathbb{n+1} \dots \mathbb{N}\mathbb{N}$. The second type of substitution is to let

$$y_1 \rightarrow 1, y_2 \rightarrow 1, \dots, y_{n-1} \rightarrow 1, y_n \rightarrow \frac{A(0, z)}{y_{n+1}y_{n+2} \dots y_N z}, \tag{25}$$

which is mostly the same as (24), except for the last step. Note that if (25) is performed on (22), the left-hand side vanishes. Based on the fact that probability generating functions are bounded for arguments lying in the unit disc and by using a similar argumentation as in [14, 15], we know that the right-hand side has to vanish as well. This provides the relation $\mathbb{n} ? \mathbb{n}\mathbb{n+1}\mathbb{n+1} \dots \mathbb{N}\mathbb{N}$.

We can arrange the substitutions of type (24) and (25) in a binary tree as shown on the left side of Fig. 2. Branches going down correspond to substitutions $y_j \rightarrow 1$, while branches to the right indicate a substitution $y_j \rightarrow A(0, z)/\tilde{y}_{j+1}z$. Hence, every path in this tree corresponds to either (24) or (25), depending on the last branch. In other words, each path represents a sequence of substitutions which, if applied to the functional equation, yield the relation indicated on the node the path ends in. Starting from the top of the tree, we can progressively determine the relations on each node, until finally, we obtain $\mathbb{N} = U(z)$ explicitly. Note that we have included the latter function in the list of unknowns notwithstanding the fact that it is known to be (6). On the right side of Fig. 2, we show

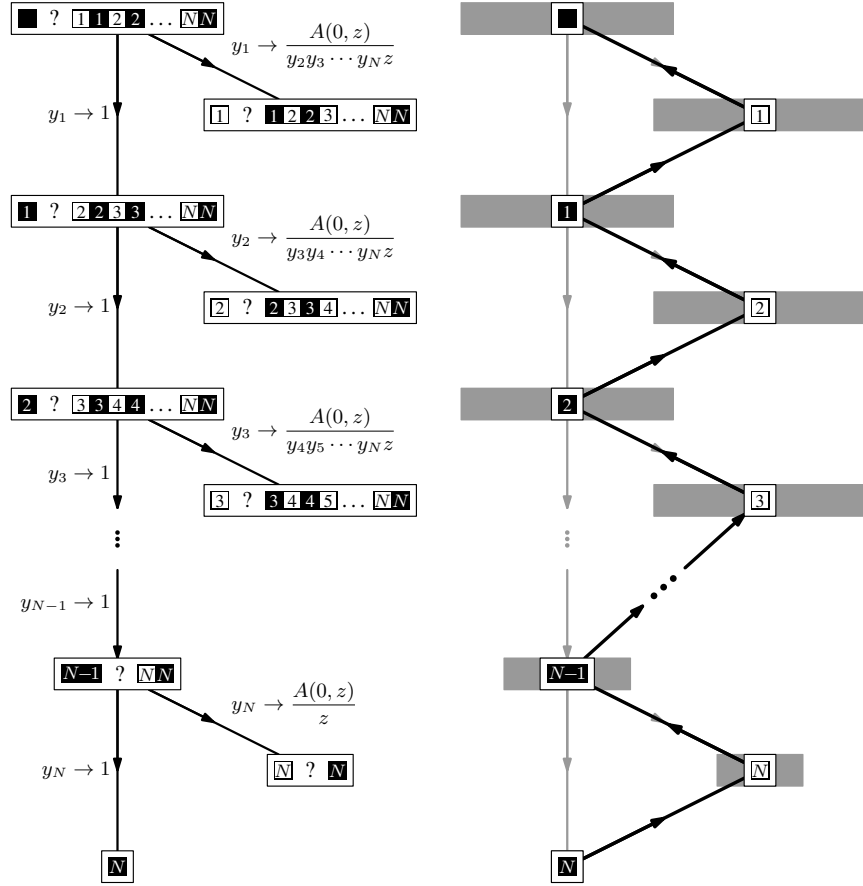


Fig. 2. Binary tree backtracking scheme to obtain the unknown functions for the MR system state distribution. First, we go from top to bottom executing the substitutions indicated on the branches. This progressively yields relations for each unknown in the list as a function of all unknowns *further* in the list. Then, it is possible to backtrack from the bottom to the top which allows to fully resolve the function on each node.

the second part of the calculation scheme. Starting from the bottom node, we work our way to the top by backtracking the previously obtained unresolved relations. Indeed, once we have \boxed{N} , we can use this in the node with relation $\boxed{N} ? \boxed{N}$ to resolve \boxed{N} . In turn, the explicit expressions \boxed{N} and \boxed{N} allow to obtain $\boxed{N-1}$ in the node with relation $\boxed{N-1} ? \boxed{N \ N}$, and so forth until we reach the top. At this point, we have an explicit expression for $\boxed{\blacksquare}$, as well as for every other unknown in the list (23). Note that in case $N = 1$, the scheme exists of only one stage containing substitutions $y_1 \rightarrow 1$ and $y_1 \rightarrow A(0, z)/z$, which we have used in [14, 15] to obtain $U(z)$ and $P(0, z)$ respectively.

We now discuss an important property regarding the behaviour of the MR system which may not readily be apparent from the analysis so far. Let us first introduce the following notation: $m_{j,k}^{[N]}$ is the position of the j th reservation at the beginning of slot k in a system with N reservations. Corresponding to (12), let also $\hat{m}_{j,k}^{[N]} = m_{j,k}^{[N]} - j$. Of course, if it is clear from the context that we are considering a system with N reservations, the superscript $[N]$ may be dropped. In what follows, we use this notation for other quantities as well, to indicate the number of reservations in the system they are related to. The following theorem is crucial to our analysis of the packet delay distribution.

Theorem 1 (Reservation Theorem). *If a queue with N reservations and a queue with $N-1$ reservations are both empty in slot 0 and are both subjected to the same number of arriving 1- and 2-packets in each of the following slots 0 to $k-1$, then we have that*

$$m_{j,k}^{[N]} = m_{j-1,k}^{[N-1]} + 1, \quad \text{or equivalently,} \quad \hat{m}_{j,k}^{[N]} = \hat{m}_{j-1,k}^{[N-1]}, \quad (26)$$

at the beginning of slot k , for $j=2, \dots, N$.

Proof. As we assume that both systems are empty in slot 0, the variables $\hat{m}_{j,0}^{[N]}$ and $\hat{m}_{j,0}^{[N-1]}$ are all equal to 0 such that (26) holds. Now, suppose that (26) holds in slot k for all $j=2, \dots, N$. If we can show that

$$\hat{m}_{j,k+1}^{[N]} = \hat{m}_{j-1,k+1}^{[N-1]}, \quad j=2, \dots, N, \quad (27)$$

then by induction, this proves the theorem. For certain values of the reservation positions in slot k , the system equations (**Empty**), (**Keep**), (**AtEnd**) and (**i -shift**) provide the new reservation positions in slot $k+1$. Therefore, we must compare these equations to their equivalent in case of a system with only $N-1$ reservations. Doing so, assuming that (26) holds, it can be checked easily that (27) holds as well, for every possible value of u_k and $a_{1,k}$. This completes the proof.

Together with the fact that the MR systems with N and $N-1$ reservations also hold the same total number of packets, theorem (26) leads to the following corollary concerning the joint pgfs of the system state of both systems. Let $P_k^{[N]}$ be the system state distribution (15) for a system with N reserved spaces. It is now easily seen that

$$\begin{aligned} P_k^{[N]}(1, y_2, \dots, y_N; z) &= E[y_2^{\hat{m}_{2,k}^{[N]}} y_3^{\hat{m}_{3,k}^{[N]}} \dots y_N^{\hat{m}_{N,k}^{[N]}} z] \\ &= E[y_2^{\hat{m}_{1,k}^{[N-1]}} y_3^{\hat{m}_{2,k}^{[N-1]}} \dots y_N^{\hat{m}_{N-1,k}^{[N-1]}} z] \\ &= P_k^{[N-1]}(y_2, \dots, y_N; z). \end{aligned} \quad (28)$$

If we let the arguments y_2 to y_n assume the value 1, then it directly follows from (28) for some $0 \leq n < N$ that

$$P_k^{[N]}(1, \dots, 1, y_{n+1}, \dots, y_N; z) = P_k^{[N-n]}(y_{n+1}, \dots, y_N; z). \quad (29)$$

This property says that the distribution of the last $N-n$ reservation positions in a system with N reservations is equal (up to a fixed shift) to that of reservation positions in a system with only $N-n$ reservations.

5 Delay analysis in multi-class queues

Before continuing, let us remind the usefulness of Little's law [21, 34], which states that $E[u] = \lambda_T E[d]$. The mean of the delay d of an arbitrary packet (regardless of type) can therefore immediately be obtained from (7). For a 2-class system, this can be written as

$$E[u] = E[v_1] + E[v_2] = \lambda_T E[d] = \lambda_1 E[d_1] + \lambda_2 E[d_2]. \quad (30)$$

Additionally, the law also applies for packets of a single type only, i.e. $E[v_i] = \lambda_i E[d_i]$, $i = 1, 2$. So for the work-conserving scheduling disciplines we consider here, once the mean delay of one type of packets is known, that of the other type directly follows from (7) and (30). That said, we now perform the per-class delay analysis ③ for the QoS-flat scheduling mechanisms. As we will see, contrary to intuition, the distributions of d_1 and d_2 are *not* necessarily the same, even though the scheduling mechanism does not distinguish between classes [50]. For FIFO, the delay of a packet of class i is $d_i = (u-1)^+ + f_i + 1$, where f_i is the number of packets that arrive in the same slot as \mathcal{P} and are stored in the queue *before* \mathcal{P} . Note that we used u instead of u_I since they have the same distribution anyway. Let a_1^I and a_2^I be the total numbers of arrivals of type 1 and 2 respectively during the arrival slot I of \mathcal{P} . Note that these variables do *not* have the same joint distribution as a_1 and a_2 . If \mathcal{P} is of type i , the joint mass function of the numbers of arrivals of type 1 and 2 during the arrival slot of \mathcal{P} is given by [5]

$$\text{Prob}[a_1^I = j_1, a_2^I = j_2] = \frac{j_i}{\lambda_i} \text{Prob}[a_1 = j_1, a_2 = j_2] \quad , \quad i = 1, 2. \quad (31)$$

Taking into account that the arrivals in slot I are stored in totally *random* order and that whenever $a_1^I + a_2^I = h$, \mathcal{P} is stored in any of the h possible positions with equal probability, one finds for the pgf $F_i(z)$ of f_i :

$$F_i(z) = \frac{1}{\lambda_i} \frac{1}{1-z} \int_z^1 \frac{\partial}{\partial z_i} A(x, x) dx, \quad \text{with } E[f_i] = \frac{\lambda_{12} + \lambda'_i}{2\lambda_i}. \quad (32)$$

Hence, the delay of a packet of type i has pgf $D_i(z) = E[z^{(u-1)^+ + f_i + 1}]$ given by

$$D_i(z) = z F_i(z) \frac{(1 - \lambda_T)(z - 1)}{z - A_T(z)}, \quad \text{with } E[d_i] = 1 + \frac{\lambda'_T}{2(1 - \lambda_T)} + \frac{\lambda_{12} + \lambda'_i}{2\lambda_i}, \quad (33)$$

the latter complying to (30). In case of LIFO, the delay of \mathcal{P} is determined by the number of packets f_i^* stored *after* \mathcal{P} in slot I instead of before. Clearly, f_i and f_i^* have the same distribution (32). In the following slots, the queue content needs to drop by f_i^* levels before \mathcal{P} can be served. We therefore have $d_i = 1 + \sum_{n=1}^{f_i^*} \eta_n$ where the η_n are iid random variables with common pgf $Y_T(z)$, known as *sub-busy periods* and defined as $\eta = \min\{h > 0 : u_{k+h} = u_k - 1\}$ for any slot k where $u_k > 0$. In other words, a sub-busy period η is the time it takes to end up with one less packet in the system as when the period started. By a

recursive probabilistic argument [5, 47], the pgf $Y_T(z)$ of η can be obtained as the solution of $Y_T(z) = zA_T(Y_T(z))$ and $Y_T(1) = 1$, from which $Y_T'(1) = \frac{1}{1-\lambda_T}$. The delay of an i -packet under LIFO is therefore

$$D_i(z) = zE[(Y_T(z))^{f_i^*}] = zF_i(Y_T(z)), \quad \text{with } E[d_i] = 1 + \frac{\lambda_{12} + \lambda_i'}{2(1-\lambda_T)\lambda_i}, \quad (34)$$

again with the latter complying to (30), as can be checked. Note that this derivation does not require knowledge of the distribution of u , as we already mentioned in section 4. For ROS, the order in which the arrivals of slot I are stored does not matter, since this mechanism does not keep any order in the queue. Both classes of packets have therefore exactly the same distribution of which the mean directly follows from (7) and (30). Higher-order moments can be derived as well, see [30].

For the per-class delay analysis of the QoS-aware mechanisms AP and MR, packets of different types arriving in slot I are not stored in the queue as a random mix so that (32) can not be used. Instead, first all 1-packets are stored and then all 2-packets. Therefore, supposing \mathcal{P} is of type i , it is useful to define ℓ_i as the number of i -packets arriving in slot I that are stored *before* (and including) \mathcal{P} . Its pgf $L_i(z)$ is found as

$$L_i(z) = \frac{z(1 - A_i(z))}{\lambda_i(1 - z)}, \quad i=1, 2, \quad (35)$$

again by considering (31) and the fact that \mathcal{P} could be *any* of the a_i^I arrivals with equal probability. Likewise, if \mathcal{P} is of type 2, the joint pgf of a_1^I and ℓ_2 is found to be

$$F(x, y) = E[x^{a_1^I} y^{\ell_2}] = \frac{y}{1-y} \frac{A_1(x) - A(x, y)}{\lambda_2}. \quad (36)$$

Now, in case of the QoS-aware absolute priority mechanism AP, the delay of 1-packets can be retrieved fairly easy. Note that the 1-packets are entirely insensitive to the presence of 2-packets. Therefore, the delay d_1 is the same as in a FIFO system where only 1-arrivals occur, i.e. where $a_2=0$ and hence $A_T(z) = A_1(z)$. For such an arrival process, (32) reduces to $F_1(z) = \frac{1-A_1(z)}{\lambda_1(1-z)}$ and (33) to

$$D_1^{\text{AP}}(z) = \frac{1-\lambda_1}{\lambda_1} z \frac{A_1(z)-1}{z-A_1(z)}. \quad (37)$$

Note that the system content distribution (6) for a FIFO-system with only 1-arrivals indeed equals the marginal distribution $U(z, 1)$ of (9), see (49). For the delay of 2-packets however, we need to take into account that d_2 is affected by 1-arrivals occurring *after* slot I [47]. We have

$$d_2 = 1 + \sum_{n=1}^{(u-1)^+ + a_1^I + \ell_2 - 1} \eta_{1,n}, \quad (38)$$

where the $\eta_{1,n}$ are iid variables denoting a sub-busy period *pertaining to 1-packets only*, i.e. the time required for $u_{1,k}$ to be reduced by one unit. Seen differently however, $\eta_{1,n}$ is also the time it takes for \mathcal{P} to advance one position in the queue. Unlike before with LIFO, now only 1-arrivals can impede the advancement of \mathcal{P} and the pgf $Y(z)$ of $\eta_{1,n}$ is therefore implicitly given by $Y(z) = zA_1(Y(z))$ with $Y(1)=1$. Hence, from (38) and using (36),

$$D_2^{\text{AP}}(z) = \frac{1-\lambda_T}{\lambda_2} z \frac{A_T(Y(z)) - A_1(Y(z))}{Y(z) - A_T(Y(z))}. \quad (39)$$

The delay moments can directly be derived from (37) and (39), as well as expressions for the tail behaviour of these distributions. Concerning the latter, we note that $D_2^{\text{AP}}(z)$ may exhibit non-exponential decay in some situations, see [37, 47] for a discussion. The delay analysis under MR of type-1 packets is given in the following section. Once $E[d_1]$ is calculated, the mean delay $E[d_2]$ can then directly be obtained from (7) and Little's law (30). However, for a *full* analysis of the delay distribution experienced by packets of type 2, we refer to our previous work [16, 17].

6 Type-1 delay of the reservation mechanism

Our purpose here is to obtain the delay d_1 experienced by a packet \mathcal{P} of type 1 as it goes through the MR system with N reserved spaces. Let us introduce

$$\omega_j = \text{Prob}[\ell_1 = j] = \frac{1}{\lambda_1} \sum_{i=j}^{+\infty} \beta_i, \quad j \geq 1, \quad (40)$$

for the mass function of ℓ_1 , such that according to (35), $L_1(z) = \sum_{j=1}^{+\infty} \omega_j z^j$. What is of importance is the exact position in which \mathcal{P} will be stored at the end of slot I . If \mathcal{P} is the first of the batch ($\ell_1 = 1$) it will seize the first reservation at position m_1 , if $\ell_1 = 2$ then it takes position m_2 , and so forth. If ℓ_1 is larger than N however, \mathcal{P} will seize a reservation created by one of the $\ell_1 - 1$ previous 1-arrivals in slot I , located somewhere at the end of the queue. We find that

$$d_1 = \begin{cases} m_j & \text{if } \ell_1 = j \leq N, \\ (u-1)^+ + \ell_1 & \text{if } \ell_1 > N. \end{cases} \quad (41)$$

Taking the z -transform and using (40), we get

$$\begin{aligned} D_1(z) &= \sum_{j=1}^N \omega_j E[z^{m_j}] + E[z^{(u-1)^+}] \sum_{j=N+1}^{+\infty} \omega_j z^j \\ &= \sum_{j=1}^N \omega_j E[z^{\hat{m}_j}] z^j + E[z^{(u-1)^+}] \left(L_1(z) - \sum_{j=1}^N \omega_j z^j \right), \end{aligned} \quad (42)$$

where $E[z^{(u-1)^+}]$ is given in (6). The marginal distribution $E[z^{\hat{m}_j}]$ appearing in (42) on the other hand, is more difficult to obtain. However, this is where Theorem 1 and its corollary (29) come into play. For $j=1, \dots, N$ we have

$$E[z^{\hat{m}_j^{[N]}}] = P^{[N]}(1, 1, \dots, 1, z, 1, 1, \dots, 1; 1) = E[z^{\hat{m}_1^{[N-j+1]}}]. \quad (43)$$

The conclusion is that instead of having to calculate the marginal distributions of all reservation positions $\hat{m}_j^{[N]}$ ($j=1, \dots, N$), it is sufficient to obtain the marginal distributions of $\hat{m}_1^{[N-j+1]}$, i.e. only of *the first reservation positions* in the systems with 1 up to N reservations.

6.1 Distribution of the first reservation position m_1

The marginal distribution of \hat{m}_1 can be obtained from the *full* system state distribution in slot I determined by functional equation (22). We derive a recursive relation for $E[z^{\hat{m}_1}]$ by using substitutions similar to those in the first stage of Fig. 2. Specifically, let all arguments in (22) be equal to 1 except for the first one, for which we take $y_1 \rightarrow z$. Using Theorem 1 again in the form of (43), we find

$$\begin{aligned} (z-\alpha)P^{[N]}(z, 1, 1, \dots, 1; 1) &= p_0(z-1)f_N(z) + \alpha(z-1)P^{[N]}(0, 1, \dots, 1; 1) \\ &+ \sum_{i=1}^{N-1} \beta_i z^i P^{[N-i]}(z, 1, 1, \dots, 1; 1), \end{aligned} \quad (44)$$

where

$$f_n(z) \triangleq \frac{1}{z-A_T(z)} \sum_{i=n}^{+\infty} \beta_i z^i, \quad n \geq 1. \quad (45)$$

Note that the factor $(z-A_T(z))^{-1}$ is entirely due to the last term of (22) where $U(z)$ appears under the mentioned substitution. Let us also define

$$\Phi_n(z) \triangleq f_n(z) - f_n(\alpha) = \frac{1}{z-A_T(z)} \sum_{i=n}^{+\infty} \beta_i z^i - \frac{1}{\alpha-A_T(\alpha)} \sum_{i=n}^{+\infty} \beta_i \alpha^i. \quad (46)$$

In (44), the probability $P(0, 1, \dots, 1; 1)$ that $\hat{m}_1 = 0$ can be obtained from the functional equation by evaluating it for the right arguments. First, let $z \rightarrow \alpha$ in (44) such that the left-hand side vanishes. As we have explained before, the other side must be equal to 0 then as well, which results in

$$P^{[N]}(0, 1, \dots, 1) = -\frac{p_0}{\alpha} f_N(\alpha) + \frac{1}{\alpha(1-\alpha)} \sum_{i=1}^{N-1} \beta_i \alpha^i P^{[N-i]}(\alpha, 1, \dots, 1). \quad (47)$$

Plugging this into (44) yields the desired recursive relation for the distribution of the first reservation position. If we first introduce a shorter notation for this

distribution, $\Omega_n(z) = P^{[n]}(z, 1, \dots, 1; 1) = \mathbb{E}[z^{\hat{m}_1^{[n]}}]$, then we finally find

$$\Omega_N(z) = p_0(z-1) \frac{\Phi_N(z)}{z-\alpha} + \sum_{i=1}^{N-1} \beta_i \frac{z^i \Omega_{N-i}(z) - \frac{z-1}{\alpha-1} \alpha^i \Omega_{N-i}(\alpha)}{z-\alpha}. \quad (48)$$

In principle, our work is done now, since this relation determines all $\Omega_n(z)$, $n=1, \dots, N$, being the distributions of the first reservation position in systems with 1 up to N reservations. Through (42) and (43), this directly provides the pgf of d_1 . Indeed, (48) can be solved in an iterative way since $\Omega_N(z)$ appears only on the left-hand side while the other side only depends on $\Omega_1(z)$ to $\Omega_{N-1}(z)$. However, the problem is that we also have to determine the constant $\Omega_n(\alpha)$ in step n ($n=1, \dots, N-1$). Since $\Omega_n(z)$ is a pgf and $\alpha < 1$, we know there must be a solution for the quantities $\Omega_n(\alpha)$ lying between 0 and 1, but obtaining these values in a direct analytic way proves to be difficult. The main issue here is that if (48) is solved by iteration, the complexity doubles with each step, producing expressions of exponentially increasing length and requiring the n -fold use of de l'Hôpital's rule. Fortunately, the quantities $\Omega_n(\alpha)$ can be obtained numerically using an entirely different approach, discussed in the next paragraph.

6.2 Obtaining the unknowns $\Omega_n(\alpha)$

It is seen that for $N = \infty$, the packets in the MR system behave the same as in the AP system. Indeed, the MR queue will decouple into two logical sub-queues in this case: one for the 1-packets at the front closest to the server and one containing a swarm of 2-packets at the far end. We call these sub-queues the 1-queue and the 2-queue respectively. In between these sub-queues there is an impenetrable barrier containing an infinite number of R's that causes a 'decoupled' operation of the system. Arrivals of type 1 will always be stored in the first R of the barrier and thus find connection to the logical 1-queue. On the other hand, the MR discipline dictates that the arriving 2-packets are stored at the end of the queue and therefore become part of the logical 2-queue. If the server becomes available, the next packet that is scheduled for service is the one positioned closest to the server. Since the 1-packets are grouped on the first positions, the server always schedules a 1-packet if one is available. Only if there are no 1-packets present, a 2-packet will jump over the barrier to be served next. We can use this resemblance of the logical 1-queue in an MR system with $N = \infty$ to the high-priority packets in the queue under AP, to obtain the limiting distribution $\Omega_\infty(z)$ of the position $\hat{m}_1^{[\infty]}$ of the first reservation. As before, note that since for AP the 1-packets are not in any way affected by the 2-packets, we know that v_1 is distributed as for FIFO given that there are no 2-arrivals. Hence, similar to (6), we have

$$U_1^{\text{AP}}(z) = (1 - \lambda_1) \frac{A_1(z)(z-1)}{z - A_1(z)}. \quad (49)$$

Now, assuming that v_1 is also the number of 1-packets in the logical 1-queue, we have

$$\hat{m}_1^{[\infty]} = m_1^{[\infty]} - 1 = (v_1 - 1)^+. \quad (50)$$

With (49), it then easily follows that

$$\Omega_\infty(z) = (1-\lambda_1) \frac{z-1}{z-A_1(z)}, \quad \text{and} \quad \Omega_\infty(\alpha) = (1-\lambda_1) \frac{\alpha-1}{\alpha-A_1(\alpha)}. \quad (51)$$

Whereas calculating the functions $\Omega_n(z)$, $n=1, 2, \dots$ iteratively from (48) is difficult, taking the transform of this sequence is much easier. Specifically, let us define $\Omega(x, z) = \sum_{n=1}^{+\infty} \Omega_n(z) x^n$, which is the generating function of the generating functions $\Omega_n(z)$. From (48) we find a closed-form expression for $\Omega(x, z)$:

$$\Omega(x, z) = \frac{p_0(z-1) \Phi(x, z) - \frac{z-1}{\alpha-1} (A_1(\alpha x) - \alpha) \Omega(x, \alpha)}{z - A_1(zx)}, \quad (52)$$

where we have used the x -transform of the functions $\Phi_n(z)$ as well, $\Phi(x, z) = \sum_{n=1}^{+\infty} \Phi_n(z) x^n$, which from (46) is calculated as

$$\Phi(x, z) = \frac{x}{1-x} \left[\frac{A_1(z) - A_1(zx)}{z - A_T(z)} - \frac{A_1(\alpha) - A_1(\alpha x)}{\alpha - A_T(\alpha)} \right]. \quad (53)$$

The function $\Omega(x, z)$, being the transform of probability generating functions, is known to be analytic for x and z lying in the unit disc. If we could find a pair (x, z) in that region for which the denominator in (52) becomes zero, then we know the numerator should be zero as well. Fortunately, one can invoke Rouché's theorem to show that if $|x| < 1$, there always exists a unique $\hat{Y}(x)$ for which $|\hat{Y}(x)| < 1$ and that satisfies

$$\hat{Y}(x) = A_1(x \hat{Y}(x)), \quad \text{with} \quad \hat{Y}(1) = 1. \quad (54)$$

Note that $\hat{Y}(z) = Y(z)/z$ where $Y(z)$ was used in (39). Now, if we let $z \rightarrow \hat{Y}(x)$ in (52), the numerator must vanish, which yields

$$\Omega(x, \alpha) = p_0(\alpha-1) \frac{\Phi(x, \hat{Y}(x))}{A_1(\alpha x) - \alpha}. \quad (55)$$

This, together with (53), allows us to write (52) as

$$\Omega(x, z) = p_0 \frac{x}{1-x} \frac{z-1}{z-A_1(zx)} \left[\frac{A_1(z) - A_1(zx)}{z - A_T(z)} - \frac{A_1(\hat{Y}(x)) - \hat{Y}(x)}{\hat{Y}(x) - A_T(\hat{Y}(x))} \right], \quad (56)$$

which determines the sequence of pgfs $\Omega_1(z), \Omega_2(z), \dots$. Let us assume a fixed (complex) value of z , then it is possible to obtain $\Omega_n(z)$ by inverting the x -transform (56). There exist many numerical methods to obtain the coefficients $[x^n] \Omega(x, z)$ of a generating function and most of them involve the evaluation of $\Omega(x, z)$ on a number of discrete points on a contour C around the origin in the x -plane. For instance, the inversion method in [1] uses a circular contour C_r of radius $0 < r < 1$ around $x = 0$. However, the problem with the evaluation of $\Omega(x, z)$ now is that the function $\hat{Y}(x)$ appearing in (56) is *not known explicitly*. Indeed, we know that $\hat{Y}(x)$ exists and is unique, but we only have the implicit

relation (54) to determine it. This complicates matters a bit, since every time we want to evaluate $\Omega(x, z)$ for a certain x on C_r (and a certain z , of course), we also have to determine $\hat{Y}(x)$ numerically from (54). To find this value, one can choose any complex root-finding algorithm to find the root $z^* = \hat{Y}(x)$ of $z^* - A_1(z^*x)$. We can apply this numerical inversion method particularly in case $z = \alpha$, i.e. to obtain the quantities $\Omega_n(\alpha)$, $n = 1, 2, \dots$ which we will need to obtain the mean value of the type-1 packet delay.

6.3 Mean value of the type-1 packet delay

For the pgf $D_1(z)$ of the type-1 packet delay, we now have from (42) and (43) that

$$D_1(z) = \sum_{j=1}^N \omega_j z^j \Omega_{N-j+1}(z) + p_0 \frac{z-1}{z-A_T(z)} \left(L_1(z) - \sum_{j=1}^N \omega_j z^j \right), \quad (57)$$

where the functions $\Omega_n(z)$ follow from the discussion in the previous paragraph. As such, $D_1(z)$ can be evaluated numerically for any particular z . The mean packet delay follows from (57) as

$$\mathbb{E}[d_1] = D'_1(1) = \sum_{j=1}^N \omega_j \Omega'_{N-j+1}(1) + L'_1(1) + \frac{\lambda'_T}{2(1-\lambda_T)} \sum_{j=N+1}^{+\infty} \omega_j, \quad (58)$$

where $\Omega'_n(1) = \mathbb{E}[\hat{m}_1^{[n]}]$, $n = 1, \dots, N$ and where we have used the fact that $p_0 = 1 - \lambda_T$. Clearly, the problem at hand is now to determine the mean value $\mathbb{E}[\hat{m}_1^{[n]}]$ of the first reservation position in a system with n reservations, $n = 1, \dots, N$. In order to do so, we assume that the quantities $\Omega_n(\alpha)$ are available. As we discussed, they can either be obtained analytically by iterating (48) and taking the limit $z \rightarrow \alpha$ in each step, or they follow from the numerical inversion method discussed in the previous paragraph. Differentiating (48) to z and taking the limit $z \rightarrow 1$ on both sides, we find after some straightforward manipulations and using (46):

$$\begin{aligned} \mathbb{E}[\hat{m}_1^{[N]}] &= \frac{1}{1-\alpha} \left[\lambda_1 - 1 + \frac{\lambda'_T}{2(1-\lambda_T)} \sum_{i=N}^{+\infty} \beta_i - \frac{p_0}{\alpha - A_T(\alpha)} \sum_{i=N}^{+\infty} \beta_i \alpha^i \right. \\ &\quad \left. + \frac{1}{1-\alpha} \sum_{i=1}^{N-1} \beta_i \alpha^i \Omega_{N-i}(\alpha) \right] + \sum_{i=1}^{N-1} \frac{\beta_i}{1-\alpha} \mathbb{E}[\hat{m}_1^{[N-i]}]. \end{aligned} \quad (59)$$

As was the case with (48), this relation can be solved iteratively. However, still assuming that we know the sequence $\Omega_n(\alpha)$, it is possible to provide a *direct* solution of the expected values $\mathbb{E}[\hat{m}_1^{[n]}]$ from (59). For the sake of clarity, let us define the following shorthands

$$\mu_n \triangleq \mathbb{E}[\hat{m}_1^{[n]}] = \Omega'_n(1), \quad n = 1, \dots, N, \quad (60)$$

$$\delta_i \triangleq \frac{\beta_i}{1-\alpha}, \quad i > 0, \quad (61)$$

$$\begin{aligned} \Gamma_n \triangleq \frac{1}{1-\alpha} & \left[\lambda_1 - 1 + \frac{\lambda'_T}{2(1-\lambda_T)} \sum_{i=n}^{+\infty} \beta_i - \frac{p_0}{\alpha - A_T(\alpha)} \sum_{i=n}^{+\infty} \beta_i \alpha^i \right. \\ & \left. + \frac{1}{1-\alpha} \sum_{i=1}^{n-1} \beta_i \alpha^i \Omega_{n-i}(\alpha) \right]. \end{aligned} \quad (62)$$

This reduces (59) to

$$\mu_N = \Gamma_N + \sum_{i=1}^{N-1} \delta_i \mu_{N-i}. \quad (63)$$

In this relation, the quantities Γ_n , $n = 1, \dots, N$ and δ_i , $i \geq 1$ are fully known whereas the quantities μ_n are the mean values we seek. One can already deduce from (63) that each μ_n will be a linear combination of the quantities Γ_1 up to Γ_n with coefficients being a function of δ_1 up to δ_n . In order to find these coefficients we proceed as follows. Let us first arrange the values obtained from (62) in a $N \times 1$ matrix $\mathbf{\Gamma}$,

$$\mathbf{\Gamma}^T \triangleq [\Gamma_1 \ \Gamma_2 \ \Gamma_3 \ \dots \ \Gamma_N]. \quad (64)$$

Secondly, we use the values (61) to define the following $N \times N$ matrix,

$$\mathbf{H} \triangleq \begin{bmatrix} \delta_1 & 1 & 0 & 0 & \dots & 0 \\ \delta_2 & 0 & 1 & 0 & \dots & 0 \\ \delta_3 & 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \ddots & \\ \delta_{N-1} & 0 & 0 & 0 & \dots & 1 \\ \delta_N & 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \quad (65)$$

This matrix \mathbf{H} is an instance of what is known as a *Leslie matrix* [31] due to P.H. Leslie who used this kind of matrices in 1945 for the study of population growth. In addition, let \mathbf{e} be the row matrix $\mathbf{e} \triangleq [1 \ 0 \ 0 \ \dots]$ of appropriate size. Now, it can be verified that μ_n is obtained by calculating the $(n-1)$ th power of \mathbf{H} , i.e. the solution of (63) is

$$\mu_n = \mathbf{e} \mathbf{H}^{n-1} \mathbf{\Gamma}, \quad n = 1, \dots, N. \quad (66)$$

This provides the mean values $\mu_n = \Omega'_n(1)$ in the expression for the mean delay (58) which now becomes

$$\mathbb{E}[d_1^{[N]}] = \mathbf{e} \left(\sum_{j=1}^N \omega_j \mathbf{H}^{N-j} \right) \mathbf{\Gamma} + \frac{\lambda'_T}{2(1-\lambda_T)} \left(1 - \sum_{j=1}^N \omega_j \right) + 1 + \frac{\lambda'_1}{2\lambda_1}. \quad (67)$$

This expression allows us to calculate the mean delay of type 1 in the system with N reservations by means of $N-1$ matrix multiplications. However, in doing so, it is possible to arrange the calculations in such a way that the mean values

$E[d_1^{[n]}]$ of the delay in the corresponding systems with *less* than N reservations are produced as well. In other words, calculating the delay in a system with one additional reservation requires only one additional matrix multiplication. The following algorithm shows how this can be achieved.

- For $n = 1, \dots, N$, calculate the values $\Omega_n(\alpha)$, either analytically or numerically, as explained before. Note that for high n , one could consider approximating $\Omega_n(\alpha)$ by the limiting value $\Omega_\infty(\alpha)$ given in (51).
- For $n = 1, \dots, N$, calculate the entries Γ_n in the matrix $\mathbf{\Gamma}$ using (62).
- Now construct the matrix \mathbf{H} as in (65) and define the starting values

$$\psi_0 = \frac{\lambda'_T}{2(1-\lambda_T)} + 1 + \frac{\lambda'_1}{2\lambda_1} \quad \text{and} \quad \mathbf{Q}_0 = \mathbf{0}. \quad (68)$$

Then, for $n = 1, \dots, N$, calculate

$$\psi_n = \psi_{n-1} - \frac{\lambda'_T}{2(1-\lambda_T)} \omega_n, \quad \text{and} \quad \mathbf{Q}_n = \mathbf{Q}_{n-1} \mathbf{H} + \omega_n \mathbf{I}, \quad (69)$$

where \mathbf{I} is the $N \times N$ identity matrix. As the mean value of $d_1^{[n]}$ follows from (67) for $N = n$, we can now see that after the n th step in this iteration, the mean delay of type 1 in a system with n reservations is given by

$$E[d_1^{[n]}] = \psi_n + \mathbf{e} \mathbf{Q}_n \mathbf{\Gamma}. \quad (70)$$

6.4 Tail distribution of the type-1 packet delay

Another important characteristic of the delay distribution besides the mean value, is its tail distribution. We use the *dominant pole approximation* which is known to yield very accurate results, see e.g. [5, 7]. Specifically, from the inversion formula for z -transforms, it follows that the probability mass function $\text{Prob}[d_1 = n]$ can be expressed as a weighted sum of negative n th powers of the poles of $D_1(z)$. Since all these poles have a modulus larger than 1, $\text{Prob}[d_1 = n]$ is dominated by the contribution of the pole z_d with the smallest modulus. It was shown that this ‘dominant’ pole z_d must necessarily be real and positive in order to ensure a nonnegative probability mass function. As such, the probability for a 1-packet to experience a delay of n slots can be expressed by the following geometric form for sufficiently large values of n :

$$\text{Prob}[d_1^{[N]} = n] \cong -\theta_1^{[N]} z_d^{-n-1}, \quad (71)$$

where z_d is the pole of $D_1(z)$ with smallest modulus and $\theta_1^{[N]}$ is the residue in z_d :

$$\theta_1^{[N]} = \text{Res}_{z_d} D_1(z) = \lim_{z \rightarrow z_d} (z - z_d) D_1(z). \quad (72)$$

The first thing to do therefore, is to identify the dominant pole z_d of $D_1(z)$. After careful inspection of the expression (57), one can prove that its dominant pole can only originate from the factor $(z - A_T(z))^{-1}$ appearing in the second term,

but also present in each $\Omega_n(z)$ through (46) and (48). Note that the multiplicity of this factor is equal to 1 in all of these terms. As such, z_d can be obtained numerically as the smallest real root larger than 1 of

$$z - A_T(z) = 0. \quad (73)$$

This value is *independent* of N and identical to the dominant pole we have in case of FIFO, see (6). Secondly, we have to evaluate the limit in (72) to obtain the residue $\theta_1^{[N]}$. Fortunately, not *all* terms in $D_1(z)$ as given in (57) have z_d as a pole. Consequently, all the contributions to $D_1(z)$ that do not, will vanish when taking the limit $z \rightarrow z_d$, due to the factor $(z - z_d)$. Therefore, if we only consider in (48) the *contributions that have a pole in z_d* , we hope that the recursion becomes easier to solve. This approach is still exact, because we only neglect terms that would vanish under the limit (72) anyway. Let us define $\Omega_n^*(z)$, $n = 1, \dots, N$ as these ‘modified’ versions of the original functions $\Omega_n(z)$ determined by (48), such that

$$\text{Res}_{z_d} \Omega_n^*(z) = \lim_{z \rightarrow z_d} (z - z_d) \Omega_n^*(z) = \lim_{z \rightarrow z_d} (z - z_d) \Omega_n(z) = \text{Res}_{z_d} \Omega_n(z). \quad (74)$$

Note that these modified functions are no longer pgfs. Their only correct interpretation is having the same residue in z_d as the original pgfs. (48) yields

$$\Omega_N^*(z) = p_0(z-1) \frac{f_N(z)}{z-\alpha} + \sum_{i=1}^{N-1} \beta_i \frac{z^i}{z-\alpha} \Omega_{N-i}^*(z), \quad (75)$$

where we recall that $f_n(z)$ is defined in (45). The required residues (74) are therefore determined by the recursion

$$\text{Res}_{z_d} \Omega_N(z) = \frac{p_0}{1 - A'_T(z_d)} \frac{z_d - 1}{z_d - \alpha} \sum_{i=N}^{+\infty} \beta_i z_d^i + \sum_{i=1}^{N-1} \beta_i \frac{z_d^i}{z_d - \alpha} \text{Res}_{z_d} \Omega_{N-i}(z), \quad (76)$$

where we have used de l'Hôpital's rule and definition (45). This relation can be represented much simpler if we introduce

$$\mu_n^* \triangleq \text{Res}_{z_d} \Omega_n(z), \quad n = 1, \dots, N, \quad (77)$$

$$\delta_i^* \triangleq \frac{\beta_i}{z_d - \alpha} z_d^i, \quad i > 0, \quad (78)$$

$$\Gamma_n^* \triangleq \frac{z_d - 1}{z_d - \alpha} \frac{p_0}{1 - A'_T(z_d)} \sum_{i=n}^{+\infty} \beta_i z_d^i, \quad n = 1, \dots, N, \quad (79)$$

similar to (60)–(62). Relation (76) then becomes

$$\mu_N^* = \Gamma_N^* + \sum_{i=1}^{N-1} \delta_i^* \mu_{N-i}^*, \quad (80)$$

which is symbolically *exactly* the same as (63) and therefore has the same kind of solution:

$$\mu_n^* = \text{Res}_{z_d} \Omega_n(z) = \mathbf{e} (\mathbf{H}^*)^{n-1} \mathbf{I}^*, \quad n=1, \dots, N. \quad (81)$$

Here, the matrix \mathbf{H}^* is the same as \mathbf{H} , but with every entry δ_i replaced by δ_i^* . Using this solution, we finally find from (57) for the residue $\theta_1^{[N]}$:

$$\theta_1^{[N]} = \mathbf{e} \left(\sum_{j=1}^N \omega_j z_d^j (\mathbf{H}^*)^{N-j} \right) \mathbf{I}^* + \frac{p_0 z_d}{\lambda_1} \frac{A_1(z_d) - 1}{1 - A'_T(z_d)} + \frac{p_0(1 - z_d)}{1 - A'_T(z_d)} \sum_{j=1}^N \omega_j z_d^j. \quad (82)$$

As with the mean value of the delay, the calculation of the residue $\theta_1^{[N]}$ can be performed in such a way that the equivalent residues $\theta_1^{[n]}$ for systems with fewer than N reservations are produced as well. The following algorithm implements this.

- For $n=1, \dots, N$, calculate the entries Γ_n^* in the matrix \mathbf{I}^* using (79).
- Now determine the values δ_i^* as in (78) and populate the matrix \mathbf{H}^* . The residues $\theta_1^{[1]}$ to $\theta_1^{[N]}$ can now progressively be obtained as follows. Define the starting values

$$\psi_0^* = p_0 \frac{z_d}{\lambda_1} \frac{A_1(z_d) - 1}{1 - A'_T(z_d)}, \quad \text{and} \quad \mathbf{Q}_0^* = \mathbf{0}. \quad (83)$$

Then, for $n=1, \dots, N$, calculate

$$\psi_n^* = \psi_{n-1}^* + p_0 \frac{1 - z_d}{1 - A'_T(z_d)} \omega_n z_d^n, \quad \text{and} \quad \mathbf{Q}_n^* = \mathbf{Q}_{n-1}^* \mathbf{H}^* + \omega_n z_d^n \mathbf{I}. \quad (84)$$

After each step, the residue $\theta_1^{[n]}$ then follows from (82) as $\theta_1^{[n]} = \psi_n^* + \mathbf{e} \mathbf{Q}_n^* \mathbf{I}^*$.

7 A comparative example

Let us consider a specific example to demonstrate the delay differentiation realised between the two packet types by MR as compared to FIFO and AP. We choose the distribution of the arrivals as

$$A(z_1, z_2) = \frac{1}{1 + \lambda_1 - z_1 \lambda_1} \cdot e^{\lambda_2(z_2 - 1)}, \quad (85)$$

i.e. the numbers of arrivals per slot of type 1 and 2 are independent and have a geometric and Poisson distribution respectively, with partial loads λ_1 and λ_2 . In Fig. 3 we plotted $E[d_1^{[N]}]$ and $E[d_2^{[N]}]$ as functions of the traffic mix λ_1/λ_T and for a fixed total load $\lambda_T = 0.9$. The mean delay under FIFO and the mean delay for 1- and 2-packets under AP are shown as well. Note that for FIFO we plotted only *one* curve, indicating the delay of an arbitrary packet regardless of its type. We see that the higher N , the more the mean delays of both types deviate from FIFO and the closer they get to their respective AP limits. On the

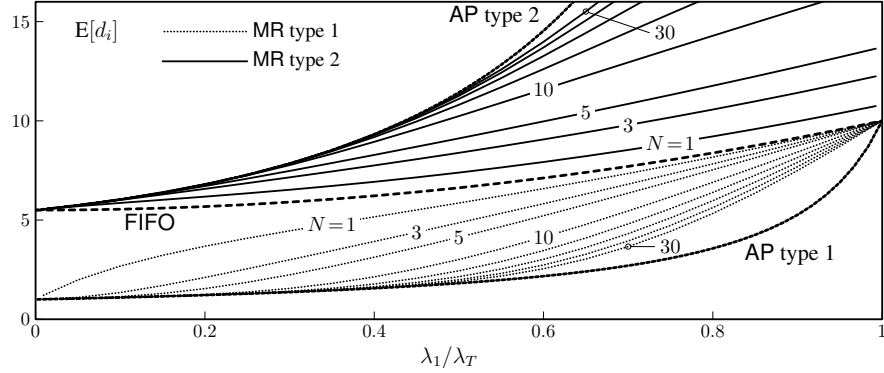


Fig. 3. Mean delay of both 1- and 2-packets versus the traffic mix λ_1/λ_T in case of $N = 1, 3, 5, 10, 15, 20, 25$ and 30 reservations in the queue. The arrivals of type 1 and type 2 are independent and have a geometric and Poisson distribution respectively with total load $\lambda_T = 0.9$.

far left side of the plot, there are but few 1-packets among a multitude of 2-packets. As a consequence, the queue contains mainly 2-packets and always has almost all of its reservations positioned directly in front of the server. Therefore, a rare arriving 1-packet can generally jump over the whole queue content and be served directly in the next slot. So even if there is only one reserved space, the behaviour under the MR mechanism is equal to that under AP for very low λ_1/λ_T , resulting in a maximal delay differentiation. On the far right of the plot, most of the traffic is of type 1, while 2-packets arrive only very rarely. From the point of view of the 1-packets, there is no difference between FIFO, AP or the intermediate MR mechanism if λ_1/λ_T is very high. However, the delay of a rare 2-packet is influenced a great deal by the queueing discipline in this case. While such a packet is almost sure to stay in the queue forever under AP, its delay under MR increases from the FIFO value more or less linearly with N . In our opinion, this is where the main strength compared to AP emerges. If only a small part of the traffic consists of low-priority traffic, their delay can be chosen at an arbitrary level by changing N , whereas the delay is almost infinite under AP (packet starvation).

8 Conclusions

We have reviewed an approach for modelling and analysing multi-class queues in heterogeneous packet network architectures (e.g. DiffServ). In particular, a discrete-time single-server queue with infinite capacity is considered. In case there are two traffic classes, type 1 with delay-sensitive and type 2 with delay-tolerant packets, we show the effectiveness of a transform domain analysis with carefully chosen supplementary variables to assess the per-class packet delay

distributions. The achieved QoS differentiation mainly depends on the scheduling mechanism in the queue. We considered, partially or in whole, First-In First-Out (FIFO), Last-In First-Out (LIFO), Random Order of Service (ROS), Absolute Priority (AP) and a novel approach based on Multiple Reservations (MR). For the latter, we discussed in detail the delay analysis of the type 1 packets in the queue.

References

1. J. Abate, W. Whitt, *Numerical Inversion of Probability Generating Functions*, *Operations Research Letters*, Vol. 12, No. 4, 1992, pp. 245–251.
2. J. Abate, W. Whitt, *Limits and Approximations for the M/G/1 LIFO Waiting-time Distribution*, *Operations Research Letters*, Vol. 20, 1997, pp. 199–206.
3. J.J. Bae, T. Suda, *Survey of Traffic Control Schemes and Protocols in ATM Networks*, *Proceedings of the IEEE*, Vol. 79, No. 2, 1991, pp. 170–189.
4. S. Blake et al., *An Architecture for Differentiated Services*, *Internet RFC 2475*, December 1998.
5. H. Bruneel, B.G. Kim, *Discrete-Time Models for Communication Systems Including ATM* (Kluwer Academic Publishers, Boston, 1993).
6. H. Bruneel, *Performance of Discrete-Time Queueing Systems*, *Computers & Operations Research*, Vol. 20, No. 3, 1993, pp. 303–320.
7. H. Bruneel, B. Steyaert, E. Desmet, G. Petit, *Analytic Derivation of Tail Probabilities for Queue Lengths and Waiting Times in ATM Multiserver Queues*, *European Journal of Operational Research*, Vol. 76, 1994, pp. 563–572.
8. W. Burakowski, H. Tarasiuk, *On New Strategy for Prioritising the Selected Flow in Queueing System*, *Proceedings of the COST 257 11th Management Committee Meeting*, (20–21 January 2000, Barcelona, Spain), COST-257 TD(00)03.
9. B.D. Choi, D.I. Choi, Y. Lee, D.K. Sung, *Priority Queueing System with Fixed-length Packet-train Arrivals*, *IEE Proceedings-Communications*, Vol. 145, No. 5, 1998, pp. 331–336.
10. L. Chuang, L. Wanming, Y. Baoping, S. Chanson, *A Dynamic Partial Buffer Sharing Scheme for Packet Loss Control in Congested Networks*, *Proceedings of the International Conference on Communication Technology*, ICCT 2000 (21–25 August 2000, Beijing, China), Vol. 2, pp. 1286–1293.
11. I. Cidon, L. Georgiadis, R. Guerin, A. Khamisy, *Optimal Buffer Sharing*, *IEEE Journal on Selected Areas in Communications*, Vol. 13, 1995, pp. 1229–1240.
12. D. Cox, *The Analysis of non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables*, *Proceedings of the Cambridge Philosophical Society*, Vol. 51, 1955, pp. 433–441.
13. A. Demers, S. Keshav, S. Shenker, *Analysis and Simulation of a Fair Queueing Algorithm*, *Proceedings of the ACM Symposium on Communications Architectures & Protocols*, SIGCOMM '89 (19–22 September 1989, Austin, TX, USA), pp. 1–12.
14. S. De Vuyst, S. Wittevrongel, H. Bruneel, *Delay Differentiation by Reserving Space in Queue*, *Electronics Letters*, 2005, Vol. 41, No. 9, pp. 69–70.
15. S. De Vuyst, S. Wittevrongel, H. Bruneel, *Place Reservation: Delay Analysis of a Novel Scheduling Mechanism*, *Computers and Operations Research*, Special Issue on 'Queues in Practice', Vol. 35, No. 8, 2008, pp. 2447–2462.

16. S. De Vuyst, S. Wittevrongel, H. Bruneel, *Parametric Delay Differentiation Between Packet Flows Using Multiple Reserved Spaces*, *Proceedings of VALUE-TOOLS 2006, the First International Conference on Performance Evaluation Methodologies and Tools* (11–13 October 2006, Pisa, Italy).
17. S. De Vuyst, S. Wittevrongel, D. Fiems, H. Bruneel, *Controlling the Delay Trade-off Between Packet Flows Using Multiple Reserved Places*, *Performance Evaluation*, Vol. 65, No. 6–7, 2008, pp. 484–511.
18. C. Dovrolis, D. Stiliadis, P. Ramanathan, *Proportional Differentiated Services: Delay Differentiation and Packet Scheduling*, *ACM Computer Communications Review*, Vol. 29, No. 4, 1999, pp. 109–120.
19. B. Feyaerts, S. De Vuyst, S. Wittevrongel, H. Bruneel, *Analysis of a Discrete-time Priority Queue with Place Reservations and Geometric Service Times*, *Proceedings of DASD 2008, the 6th Symposium on Design, Analysis and Simulation of Distributed Systems* (16–19 June 2008, Edinburgh, UK), pp. 140–147.
20. B. Feyaerts, S. Wittevrongel, *Performance Analysis of a Priority Queue with Place Reservation and General Transmission Times*, *Proceedings of EPEW 2008, the 5th European Performance Engineering Workshop* (24–25 September 2008, Palma de Mallorca, Spain), *Lecture Notes in Computer Science*, Vol. 5261, pp. 197–211.
21. D. Fiems, H. Bruneel, *A Note on the Discretization of Little's Result*, *Operations Research Letters*, Vol. 30, No. 1, 2002, pp. 17–18.
22. L. Flatto, *The Waiting Time Distribution for the Random Order Service M/M/1 Queue*, *Annals of Applied Probability*, 1997, Vol. 7, pp. 382–409.
23. S. Floyd, V. Jacobson, *Random Early Detection Gateways for Congestion Avoidance*, *IEEE-ACM Transactions on Networking*, Vol. 1, No. 4, August 1993, pp. 397–413.
24. H.R. Gail, S.L. Hantler, B.A. Taylor, *On a Preemptive Markovian Queue with multiple servers and two priority classes*, *Mathematics of Operations Research*, Vol. 17, No. 2, 1992, pp. 365–391.
25. P. Gevros, J. Crowcroft, P. Kirstein, S. Bhatti, *Congestion Control Mechanisms and the Best Effort Service Model*, *IEEE Network*, May-June 2001, pp. 16–26.
26. P. Hurley, J.-Y. Le Boudec, P. Thiran, M. Kara, *ABE: Providing a Low-Delay Service within Best Effort*, *IEEE Network*, May-June 2001, pp. 60–69.
27. S. Kausha, R.K. Sharma, *Modeling and Analysis of Adaptive Buffer Sharing Scheme for Consecutive Packet Loss Reduction in Broadband Networks*, *International Journal of Computer Systems Science and Engineering*, Vol. 4, No. 1, 2007, pp. 8–15.
28. L. Kosten, *Stochastic Theory of a Multi-entry Buffer (1)*, *Delft Progress Report*, 1974, Vol. 1, pp. 10–18.
29. K. Kontovasilis, S. Wittevrongel, H. Bruneel, B. Van Houdt, C. Blondia, *Performance of Telecommunication Systems: Selected Topics*, *Proceedings of the 17th IFIP World Computer Congress*, (Montreal, 25–30 August 2002), pp. 61–93.
30. K. Laevens, H. Bruneel, *Delay Analysis for ATM Queues with Random Order of Service*, *Electronic Letters*, Vol. 31, No. 5, 1995, pp. 346–347.
31. P.H. Leslie, *On the Use of Matrices in Certain Population Mathematics*, *Biometrika*, Vol. 33, 1945, pp. 183–212.
32. Y. Lim, J.E. Kobza, *Analysis of a Delay-dependent Priority Discipline in an Integrated Multiclass Traffic Fast Packet Switch*, *IEEE Transactions on Communications*, Vol. 38, No. 5, 1990, pp. 659–685.
33. K.Y. Liu, D.W. Petr, V.S. Frost, H.B. Zhu, C. Braun, W.L. Edwards, *Design and Analysis of a Bandwidth Management Framework for ATM-Based Broadband ISDN*, *IEEE Communications Magazine*, Vol. 35, No. 5, 1997, pp. 138–145.

34. J.D.C. Little, A Proof of the Queuing Formula $L = \lambda W$, *Operations Research*, Vol. 9, No. 3, 1961, pp. 383–387.
35. T. Maertens, J. Walraevens, H. Bruneel, On Priority Queues with Priority Jumps, *Performance Evaluation*, Vol. 63, 2006, pp. 1235–1252.
36. T. Maertens, J. Walraevens, H. Bruneel, Performance Comparison of Several Priority Schemes with Priority Jumps, *Annals of Operations Research*, Vol. 162, 2008, pp. 109–125.
37. T. Maertens, J. Walraevens, H. Bruneel, Non-exponential Tail Probabilities in Queuing Systems, *Proceedings of ITC-19, 19th International Teletraffic Congress* (29 August–2 September 2005, Beijing, China), pp. 1155–1164.
38. M. Menth, M. Schmid, H. Heiss, T. Reim, MEDF – A Simple Scheduling Algorithm for Two Real-time Transport Service Classes with Application in the UTRAN, *Proceedings of INFOCOM '03* (30 March – 3 April 2003, San Francisco, USA).
39. P. Van Mieghem, B. Steyaert, G.H. Petit, Performance of Cell Loss Priority Management Schemes in a Single Server Queue, *International Journal of Communication Systems*, Vol. 10, 1997, pp. 161–180.
40. M. Mowbray, G. Karlsson, T. Köhler, Capacity Reservation for Multimedia Traffic, *Distrib. Syst. Engng.*, Vol. 5, 1998, pp. 12–18.
41. A.K. Parekh, R.G. Gallager, A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case, *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3, June 1993, pp. 344–357.
42. Semeria C., Supporting Differentiated Service Classes: Queue Scheduling Disciplines, *Juniper Networks, White Paper*, 2001.
43. V. Sivaraman, F. Chiussi, Providing End-to-End Statistical Delay Guarantees with Earliest Deadline First Scheduling and Per-Hop Traffic Shaping, *Proceedings of INFOCOM 2000* (26–30 March 2000, Tel Aviv, Israel).
44. A. Striegel, G. Manimaran, Packet Scheduling with Delay and Loss Differentiation, *Computer Communications*, Vol. 25, 2000, pp. 21–31.
45. T. Takine, B. Sengupta, T. Hasegawa, An Analysis of a Discrete-Time Queue for Broadband ISDN with Priorities Among Traffic Classes, *IEEE Transactions on Communications*, Vol. 42, No. 2–4, 1994, pp. 1837–1845.
46. C.-K. Tham, Q. Yao, Y. Jian, A Multi-class Probabilistic Priority Scheduling Discipline for Differentiated Services Networks, *Computer Communications*, Vol. 25, 2002, pp. 1487–1496.
47. J. Walraevens, B. Steyaert, H. Bruneel, Performance Analysis of a Single-Server ATM Queue with a Priority Scheduling, *Computers & Operations Research*, Vol. 30, No. 12, 2003, pp. 1807–1829.
48. J. Walraevens, S. Wittevrongel, H. Bruneel, A Discrete-Time Priority Queue with Train Arrivals, *Stochastic Models*, Vol. 23, No. 3, 2007, pp. 489–512.
49. S. Wittevrongel, H. Bruneel, Discrete-time ATM Queues with Independent and Correlated Arrival Streams, *Chapter 16 in: Performance Evaluation and Applications of ATM Networks* (Kluwer Academic Publishers, Boston, 2000), pp. 387–412.
50. S. Wittevrongel, H. Bruneel, Per-source Mean Cell Delay and Mean Buffer Contents in ATM Queues, *Electronics Letters*, Vol. 33, No. 6, 1997, pp. 461–462.
51. S.F. Yashkov, Processor Sharing Queues: Some Progress in Analysis, *Queueing Systems*, Vol. 2, 1987, pp. 1–17.
52. L. Zhang, Virtual Clocks: a New Traffic Control Algorithm for Packet Switching Networks, *Proceedings of the ACM Symposium on Communications Architectures & Protocols, SIGCOMM '90* (24–27 September 1990, Philadelphia, PA, USA), pp. 19–29.